Attorney Docket No. UK999-027

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

I hereby certify that this paper is being deposited on this date with the U.S. Postal Service as first class mail addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Signature: _____ Date: March 21, 2006

Applicant(s): H. E. Butterworth et al.
Docket No.: UK999-027
Serial No.: 09/401,676
Filing Date: September 22, 1999
Group: 2131
Examiner: Christian A. La Forgia

Title: Data Processing Systems and Method for Processing Work Items in Such Systems

---

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313

Sir:

Applicants (hereinafter referred to as "Appellants") hereby appeal the final rejection of claims 1-14 of the above referenced application.

REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corp., as evidenced by an assignment recorded September 22, 1999 in the U.S. Patent and Trademark Office at Reel 10279, Frame 0510. The assignee, International Business Machines Corp., is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no known related appeals and interferences.

STATUS OF CLAIMS

Claims 1-14 are pending in the present application. Claims 1-3, 5-7 and 9-11 stand rejected under 35 U.S.C. §103(a) and claims 12-14 stand rejected under 35 U.S.C. §102(a), (e) and nonstatutory double patenting. Claims 1-14 are appealed.

STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

The present invention relates generally to processing of work items in data processing systems and more particularly to the scheduling of tasks to process such work items (Specification, page 1, lines 5-8).

By way of example, as recited in claim 1, a method for processing work items in a data processing system comprises the following steps. An interrupt is generated in response to receipt of a work item in the system. System interrupts are disabled. A task is scheduled through the generated interrupt for processing of the work item. The task is executed to process the work item. Additional work items received by the system are processed. When there are no additional work items for processing, a further task is speculatively scheduled for processing of subsequently received work items in the system, without enabling system interrupts.

As a further example, as recited in claim 2, a method for processing work items in a data processing system further comprises the following steps. The speculatively scheduled task is executed to process work items received by the system. System interrupts are enabled when no additional work items have been received by the system when the speculatively scheduled task is executed. One or more work items are processed when at least one work item has been received by the system when the speculatively scheduled task is executed. An additional further task is speculatively scheduled for processing of subsequently received work items after processing the one or more work items, without enabling system interrupts.

By way of example, illustrative embodiments of the invention of claims 1 and 2 are shown in FIG. 2 of the drawings. FIG. 2 is a flow diagram showing the steps involved in a method according to a preferred embodiment of the invention. At block 320, an interrupt is generated, and

at block 330, system interrupts are disabled. At block 340, a task is scheduled, and at block 350, the task is executed. At block 360, work items are processed, and when there are no further work items, a task is speculatively scheduled at block 380. At block 390, the speculatively scheduled task is executed. When there are no work items, interrupts are enabled at block 310. When there are work items, they are processed at block 410, and a task is speculatively scheduled at block 380 when there are no further work items to process. Schematic representations of the state of the task and work item queues of the preferred embodiment of the invention at different states of the method of FIG. 2 are shown in FIGS. 3A, 3B and 3C.

In accordance with one embodiment of the invention, the method could include the step of continually scheduling speculative tasks (i.e. polling) for processing of work items that may subsequently be received in the system (Specification, page 4, lines 8-16). In a preferred method, when the speculatively scheduled task is executed to process any work items received by the system and it is determined that there are no work items, the interrupt is enabled. Thus, when the system is fully utilized, the interrupt mechanism is replaced with a polling mechanism involving a continuous series of speculatively scheduled tasks. However, when the system or device utilization decreases (i.e., when there are no work items when the speculatively scheduled task is processed), then the system reverts to interrupts (Specification, page 4, lines 18-27).

Finally, as recited in claim 12, a method for processing work items in a data processing system, comprises the following steps. An interrupt-based mechanism is effectively provided for processing work items when system utilization is low with respect to work items. A polling-based mechanism is effectively provided for processing work items when system utilization is relatively high with respect to work items.

The above-mentioned claims and figures describe a process, operable on a storage controller, for processing work items from the host system in a manner that combines the best attributes of the polling and interrupt methods to service the host work items with low latency at low utilization and by polling for low overhead at high utilization (Specification, page 10, lines 2-8).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. Claims 1-3, 5-7 and 9-11 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,414,858 to Hoffman et al. (hereinafter "Hoffman") in view of Applicant's Admitted Prior Art (hereinafter "AAPA").

II. Claims 12-14 stand rejected under 35 U.S.C. §102(a) and (e) as being anticipated by U.S. Patent No. 5,933,598 to Scales et al. (hereinafter "Scales").

ARGUMENT

Appellants incorporate by reference herein the disclosure of all previous responses filed in the present application, namely, responses dated January 24, 2003, July 3, 2003, September 8, 2003, May 3, 2004, August 18, 2005, and December 28, 2005. Sections I and II to follow will respectively address grounds I and II presented above.

Appellants initially note that the status of claims 4 and 8 is unclear. These claims are only addressed in paragraph 11 of the final Office Action where the Examiner merely restates the claims and contends the elements are discussed in AAPA. Appellants strongly disagree with this contention.

I. Obviousness rejection over Hoffman in view of AAPA

With regard to the rejection of claims 1-3, 5-7 and 9-11 under 35 U.S.C. §103(a) as being unpatentable over Hoffman in view of AAPA, Appellants respectfully assert that the cited combination fails to establish a prima facie case of obviousness under 35 U.S.C. §103(a), as specified in M.P.E.P. §2143.

As set forth therein, M.P.E.P. §2143 states that three requirements must be met to establish a prima facie case of obviousness. For example, the cited combination must teach or suggest all the claim limitations. Thus, it is sufficient to show that a prima facie case of obviousness has not been established by showing that this requirement has not been met.

A. Claims 1, 3, 5, 7, 9 and 10

The collective teaching of Hoffman and AAPA fails to suggest or render obvious at least the elements of independent claims 1, 5 and 10 of the present invention. For at least this reason, a prima facie case of obviousness has not been established.

The present invention, as recited in independent claim 1, recites a method of processing work items in a data processing system, comprising the steps of: (i) generating an interrupt in response to receipt of a work item in the system; (ii) disabling system interrupts; (iii) scheduling a task through the generated interrupt for processing of the work item; (iv) executing the task to process the work item; (v) processing additional work items received by the system; and (vi) when there are no additional work items for processing, speculatively scheduling a further task for processing of subsequently received work items in the system, without enabling system interrupts. Independent claims 5 and 10 recite other aspects of the invention comprising similar limitations.

Hoffman discloses a system and method for dynamically varying between interrupt and polling methods to service requests of computer peripherals. The rates of incoming requests are tracked, and if a rate meets a specified threshold, the method may transition from interrupt to polling, or polling to interrupt. AAPA simply discloses that during the handling of a service request through an interrupt, system interrupts are disabled, then reenabled when the handling of the service request is complete.

Independent claims 1, 5 and 10 describe a transition from an interrupt method to a polling method. This transition takes place after a single interrupt is received by the system, through the disabling of system interrupts and scheduling, executing and processing steps of claim 1. When there are no additional work items for processing in the polling method, a further task is speculatively scheduled for processing of subsequently received work items in the system, without enabling system interrupts, as recited in independent claims 1, 5 and 10.

While Hoffman discloses a transition between interrupt and polling methods when a certain request rate is reached, the transition of the present invention is based on the reception of a single interrupt. Additionally, Hoffman fails to disclose the speculative scheduling of a further task for processing of subsequently received work items, when there are no additional work items for processing.

AAPA fails to remedy the deficiencies described above with regard to Hoffman in that AAPA provides no discussion of speculatively scheduled tasks. Therefore, the combination of Hoffman and AAPA fails to suggest or render obvious the elements of independent claims 1, 5 and 10.

Dependent claims 3, 7 and 9 are patentable at least by virtue of their respective dependency from independent claims 1 and 5, and also recite patentable subject matter in their own right.

B. Claims 2, 6 and 11

Dependent claims 2, 6 and 11 are patentable at least by virtue of their respective dependency from independent claims 1, 5 and 10, and also recite patentable subject matter in their own right.

The collective teaching of Hoffman and AAPA fails to suggest or render obvious the elements of dependent claims 2, 6 and 11 of the present invention. For at least this reason, a prima facie case of obviousness has not been established.

The present invention, as recited in dependent claim 2, recites further steps in the method of processing work items in a data processing system, comprising: (vii) executing the speculatively scheduled task to process work items received by the system; (viii) enabling system interrupts when no additional work items have been received by the system when the speculatively scheduled task is executed; (ix) processing one or more work items when at least one work item has been received by the system when the speculatively scheduled task is executed; and (x) speculatively scheduling an additional further task for processing of subsequently received work items after processing the one or more work items, without enabling system interrupts. Dependent claims 6 and 11 recite other aspects of the invention comprising similar limitations.

Dependent claims 2, 6 and 11 recite the possible transition from the polling method to the interrupt method. While Hoffman discloses a transition from a polling method to an interrupt method when a certain request rate is reached, the transition of the present invention is based on a previously speculatively scheduled task finding no additional work items received by the system for processing. Additionally, Hoffman fails to disclose the processing of one or more received work items when the speculatively scheduled task is executed, or the speculative scheduling of an additional further task for processing of subsequently received work items after processing the received work items.

As discussed above, AAPA provides no discussion regarding speculatively scheduled tasks, and thus fails to remedy the deficiencies of Hoffman described above with regard to claims 2, 6 and 11. Therefore, the combination of Hoffman and AAPA fails to suggest or render obvious the elements of claims 2, 6 and 11.

Accordingly, withdrawal of the rejection to claims 1-3, 5-7, and 9-11 under 35 U.S.C. §103(a) is therefore respectfully requested.

II. Anticipation Rejection over Scales

A. Claims 12-14

With regard to the rejection of claims 12-14 under 35 U.S.C. §102(a) and (e) as being anticipated by Scales, Appellants respectfully assert that Scales fails to teach or suggest all of the limitations in claims 12-14 for at least the reasons presented in Appellants' previous responses as well as the reasons presented below.

It is well-established law that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). Appellants assert that the rejection based on Scales does not meet this basic legal requirement. Support for this assertion follows.

The present invention, for example, as recited in independent claim 12, recites a method of processing work items where an interrupt-based mechanism for processing work items is provided when system utilization is low with respect to work items, and a polling-based mechanism for processing work items is provided when system utilization is relatively high with respect to work items.

Scales discloses methods for enabling data sharing among workstations of a distributed shared memory system using variable sized quantities of data. In particular, the portion of Scales cited by the Examiner discloses a polling mechanism used to process messages generated by the workstations, and the advantages of such a mechanism over an interrupt mechanism.

Independent claim 12 of the present invention differs from Scales in that it discloses the use of an interrupt based-mechanism and a polling-based mechanism. While Scales discloses a polling mechanism, and the possible use of an interrupt mechanism instead of the polling mechanism, it

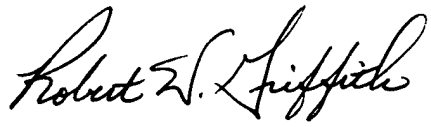
does not disclose using both a polling mechanism and an interrupt mechanism. Further, Scales does not disclose mechanisms that are dependent on utilization with respect to work items as recited in claim 12 of the present invention.

Dependent claims 13 and 14 are patentable at least by virtue of their dependency from independent claim 12, and also recite patentable subject matter in their own right. Accordingly, withdrawal of the rejection to claims 12-14 under 35 U.S.C. §102(a) and (e) is therefore respectfully requested.

Appellants reserve the right to file a terminal disclaimer to overcome the double patenting rejection of claims 12-14.

In view of the above, Appellants believe that claims 1-14 are in condition for allowance, and respectfully request withdrawal of the §103(a), §102(a) and §102(e) rejections.

Respectfully submitted,



Date: March 21, 2006

Robert W. Griffith
Attorney for Applicant(s)
Reg. No. 48,956
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-4547

CLAIMS APPENDIX

1. A method of processing work items in a data processing system comprising the steps of:

generating an interrupt in response to receipt of a work item in the system;

disabling system interrupts;

scheduling a task through the generated interrupt for processing of the work item;

executing the task to process the work item;

processing additional work items received by the system; and

when there are no additional work items for processing, speculatively scheduling a further task for processing of subsequently received work items in the system, without enabling system interrupts.

2. The method of claim 1, further comprising the steps of:

executing the speculatively scheduled task to process work items received by the system;

enabling system interrupts when no additional work items have been received by the system when the speculatively scheduled task is executed;

processing one or more work items when at least one work item has been received by the system when the speculatively scheduled task is executed; and

speculatively scheduling an additional further task for processing of subsequently received work items after processing the one or more work items, without enabling system interrupts.

3. The method of claim 1, wherein the work items are managed on a queue.

4. The method of claim 1, wherein the event that further work items are received after the task is scheduled and prior to execution of the task, the step of executing the task comprises processing all the received work items.

5. A data processing system comprising:

processing means for executing tasks to process work items in the data processing system; and interrupt generating means for generating an interrupt in response to receipt of a work item in the system; wherein the processing means is operable to:

disable system interrupts;

schedule a task through the generated interrupt for processing of the work item;

execute the task to process the work item;

process additional work items received by the system; and

when there are no additional work items for processing, speculatively schedule a further task for processing of subsequently received work items in the system, without enabling system interrupts.

6. The data processing system of claim 5, the processing means being operable to execute the speculatively scheduled task to process work items received by the system, enable system interrupts when no additional work items have been received by the system when the speculatively scheduled task is executed, process one or more work items when at least one work item has been received by the system when the speculatively scheduled task is executed, and speculatively schedule an additional further task for processing of subsequently received work items after processing the one or more work items, without enabling system interrupts.

7. The data processing system of claim 5, further comprising memory for storing the received work items on a queue.

8. The data processing system of claim 5, wherein the event that further work items are received after the task is scheduled and prior to execution of the task, the processing means is operable to execute the task to process all the work items.

9. The data processing system of claim 5, wherein the interrupt generating means and processing means are embodied in a data storage controller and the work items comprise data transfer requests from an attached host system.

10. A computer program product comprising a computer usable medium having computer readable program code means embodied in the medium for processing work items in a data processing system, the program code means comprising:

code means for causing the data processing system to generate an interrupt in response to receipt of a work item in the system;

code means for causing the data processing system to disable system interrupts;

code means for causing the data processing system to schedule a task through the generated interrupt for processing of the work item;

code means for causing the data processing system to execute the task to process the work item;

code means for causing the data processing system to process additional work items received by the system; and

code means for causing the data processing system to speculatively schedule a further task for processing of subsequently received work items in the system when there are no additional work items for processing, without enabling system interrupts.

11. The computer program product of claim 10, the computer readable program code means further comprising:

code means for causing the data processing system to execute the speculatively scheduled task to process work items received by the system;

code means for causing the data processing system to enable system interrupts when no additional work items have been received by the system when the speculatively scheduled task is executed;

code means for causing the data processing system to process one or more work items when at least one work items has been received by the system when the speculatively scheduled task is executed; and

code means for causing the data processing system to speculatively schedule an additional further task for processing of subsequently received work items after processing the one or more work items, without enabling system interrupts.

12. A method of processing work items in a data processing system, comprising:

effectively providing an interrupt-based mechanism for processing work items, when system utilization is low with respect to work items; and

effectively providing a polling-based mechanism for processing work items, when system utilization is relatively high with respect to work items.

13. A method as claimed in claim 12 wherein work items are received in accordance with at least one device driver associated with a host system.

14. A method as claimed in claim 12 wherein the data processing system comprises a storage controller.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.